ACAIE Implementation Details

Druv Pai

Contents

1 Quick Start

1.1 Installation Guide

The following steps provide a method for quickly and properly installing the simulation program as well as its dependencies. This method should hopefully be platform-agnostic.

1. Open and unzip the software into a folder of your choice.
2. Download and install, through a package manager or otherwise, the Anaconda distribution for Python 3.6 or later. The software should install the requisite version of Python if it's not already on the system, but if it doesn't for some reason, download and install it as well.
3. Ensure that Anaconda is in your `PATH` environment variable. Write and run the following command in a terminal, such as the Unix terminal on Mac or Linux systems or Git Bash for Windows systems:

   ```
   conda
   ```

   If your terminal doesn't fail to recognize the `conda` executable, move on. If it does fail to recognize the program, retry the installation, or otherwise safely add the `conda` executable to the `PATH` environment variable.
4. Navigate, in the terminal, to the folder containing the software and ensure that it contains a file named `environment.yml`. Write and run the following command in terminal:

   ```
   conda env create -f environment.yml
   ```

5. Ensure that the environment has been created successfully. Write and run the following command in terminal:

   ```
   conda-env list
   ```

   If an environment named `ACAIE` is displayed, then the procedure has succeeded. Else, try again from the beginning or wherever the procedure may have failed.

If all this has been done, the program is effectively installed.

1.2 Usage Guide

The following steps are to be done whenever running the simulation program. It will ensure that the program has all its dependencies, by loading into the Anaconda environment packaged with the software.

1. Navigate, in the terminal, to the folder containing the software. In the terminal, write and run the following command:

   `activate ACAIE` (on Windows), `source activate ACAIE` (on MacOS or Linux)

   If this is succesful, the text `(ACAIE)` will appear on the next lines of text in the terminal before the typable-in field. This will appear until the command

   `deactivate` (on Windows), `source deactivate` (on MacOS or Linux)

   is written and run.
2. Execute the program. Write and run the command

   `jupyter notebook`

   in terminal. Select `simulation.ipynb`. Follow the instructions on the Jupyter Notebook.

2 Program Components

Here we discuss a high level organization of the system. This is here for the purpose of program organization and to allow easy extensions to the program.

1. `simulation.ipynb`: the driver notebook that allows for human interaction with the program, as well as provides an interface for plotting.
2. `system.py`: the file which holds the `ACAIE` class and simulation and plot functions.
3. `dae.py`: the file which holds the methods used to integrate the differential algebraic equation (DAE) modeling the chemical system. The integration scheme utilizes the Assimulo package.
4. `constants.py`: the file that holds all the constants and some auxiliary functions, namely to determine some "constants" which actually vary with pH, as well as compute an adsorbed concentration vector.

3 Equations and Constants

3.1 Equations

The main system simulated is given by the following equations, which are obtained from the article *Modeling* As(III) *Oxidation and Removal with Iron Electrocoagulation in Groundwater*.

First, there is a species mass balance:

$$[\mathrm{As(III)}]_0 + [\mathrm{As(V)}]_0 = [\mathrm{As(III)}] + [\mathrm{As(III)}]_{\mathrm{ads}} + [\mathrm{As(V)}]_{\mathrm{tot}} \tag{1}$$

$$[\mathrm{As(V)}]_{\mathrm{tot}} = [\mathrm{As(V)}]_{\mathrm{ads}} + [\mathrm{As(V)}] \tag{2}$$

$$[\mathrm{P}] = [\mathrm{P}]_0 - [\mathrm{P}]_{\mathrm{ads}} \tag{3}$$

$$[\mathrm{Si}] = [\mathrm{Si}]_0 - [\mathrm{Si}]_{\mathrm{ads}} \tag{4}$$

There are also some kinetics equations:

$$\frac{\mathrm{d}[\mathrm{Fe(II)}]}{\mathrm{d}t} = -k_{\mathrm{app}}[\mathrm{Fe(II)}][\mathrm{O_2}] + \frac{\mathrm{d}[\mathrm{Fe(II)}]_{\mathrm{dosage}}}{\mathrm{d}t} - k_{\mathrm{H_2O_2}}[\mathrm{Fe(II)}][\mathrm{H_2O_2}] \tag{5}$$

$$\frac{\mathrm{d}[\mathrm{Fe(III)}]}{\mathrm{d}t} = k_{\mathrm{app}}[\mathrm{Fe(II)}][\mathrm{O_2}] + k_{\mathrm{H_2O_2}}[\mathrm{Fe(II)}][\mathrm{H_2O_2}] \tag{6}$$

$$\frac{\mathrm{d}[\mathrm{O_2}]}{\mathrm{d}t} = k_\mathrm{r}([\mathrm{O_2}]_\mathrm{sat} - [\mathrm{O_2}]) - k_\mathrm{app}[\mathrm{Fe(II)}][\mathrm{O_2}] \tag{7}$$

$$\frac{\mathrm{d}[\mathrm{As(V)}]_\mathrm{tot}}{\mathrm{d}t} = \frac{\beta}{1 + \frac{k_1[\mathrm{Fe(II)}]}{k_2[\mathrm{As(III)}]}} k_\mathrm{app}[\mathrm{Fe(II)}][\mathrm{O_2}] \tag{8}$$

$$\frac{\mathrm{d}[\mathrm{H_2O_2}]}{\mathrm{d}t} = \frac{\mathrm{d}[\mathrm{Fe(II)}]_\mathrm{dosage}}{\mathrm{d}t} - k_{\mathrm{H_2O_2}}[\mathrm{Fe(II)}][\mathrm{H_2O_2}] - L(t) \approx \frac{\mathrm{d}[\mathrm{Fe(II)}]_\mathrm{dosage}}{\mathrm{d}t} - k_{\mathrm{H_2O_2}}[\mathrm{Fe(II)}][\mathrm{H_2O_2}] \tag{9}$$

$$\frac{\mathrm{d}[\mathrm{H_2}]}{\mathrm{d}t} = 2\epsilon_\mathrm{red} \frac{\mathrm{d}[\mathrm{Fe(II)}]_\mathrm{dosage}}{\mathrm{d}t} \approx 0\,\mathrm{M\,s^{-1}} \tag{10}$$

The above equations will be used in the form of the differential algebraic equation.

Also, there are some adsorption equations:

$$[\mathrm{As(III)}]_\mathrm{ads} = \frac{q_\mathrm{max}[\mathrm{Fe(III)}]K_\mathrm{As(III)}[\mathrm{As(III)}]}{1 + K_\mathrm{As(III)}\mathrm{As(III)} + K_\mathrm{As(V)}[\mathrm{As(V)}] + K_\mathrm{P}[\mathrm{P}] + K_\mathrm{Si}[\mathrm{Si}]}$$

$$[\mathrm{As(V)}]_\mathrm{ads} = \frac{q_\mathrm{max}[\mathrm{Fe(III)}]K_\mathrm{As(V)}[\mathrm{As(V)}]}{1 + K_\mathrm{As(III)}\mathrm{As(III)} + K_\mathrm{As(V)}[\mathrm{As(V)}] + K_\mathrm{P}[\mathrm{P}] + K_\mathrm{Si}[\mathrm{Si}]}$$

$$[\mathrm{P}]_\mathrm{ads} = \frac{q_\mathrm{max}[\mathrm{Fe(III)}]K_\mathrm{P}[\mathrm{P}]}{1 + K_\mathrm{As(III)}\mathrm{As(III)} + K_\mathrm{As(V)}[\mathrm{As(V)}] + K_\mathrm{P}[\mathrm{P}] + K_\mathrm{Si}[\mathrm{Si}]}$$

$$[\mathrm{Si}]_\mathrm{ads} = \frac{q_\mathrm{max}[\mathrm{Fe(III)}]K_\mathrm{Si}[\mathrm{Si}]}{1 + K_\mathrm{As(III)}\mathrm{As(III)} + K_\mathrm{As(V)}[\mathrm{As(V)}] + K_\mathrm{P}[\mathrm{P}] + K_\mathrm{Si}[\mathrm{Si}]}$$

Here we compute some initial conditions which are absent from the formulation of the differential algebraic equation. Note that, since

$$\frac{\mathrm{d}[\mathrm{As(III)}]}{\mathrm{d}t} = -k_2[\mathrm{As(III)}][\mathrm{Fe(IV)}]$$

and $[\mathrm{Fe(IV)}] \approx 0\,\mathrm{M\,s^{-1}}$ at $t = 0$ since it's not part of the original system,

$$\left.\frac{\mathrm{d}[\mathrm{As(III)}]}{\mathrm{d}t}\right|_{t=0} \approx 0\,\mathrm{M\,s^{-1}}$$

Also, through more complicated but similar calculations using the mass balance and adsorbance it can be shown that

$$\left.\frac{\mathrm{d}[\mathrm{As(V)}]}{\mathrm{d}t}\right|_{t=0} \approx \left.\frac{\mathrm{d}[\mathrm{P}]}{\mathrm{d}t}\right|_{t=0} \approx \left.\frac{\mathrm{d}[\mathrm{Si}]}{\mathrm{d}t}\right|_{t=0} \approx 0\,\mathrm{M\,s^{-1}}$$

The rest can be computed directly from the kinetic equations given above.

Define the vector

$$\mathbf{c} = \begin{bmatrix} [\mathrm{As(III)}] \\ [\mathrm{As(V)}] \\ [\mathrm{P}] \\ [\mathrm{Si}] \\ [\mathrm{Fe(II)}] \\ [\mathrm{Fe(III)}] \\ [\mathrm{O_2}] \\ [\mathrm{As(V)}]_\mathrm{tot} \\ \mathrm{pH} \end{bmatrix}$$

Then we have the equation

$$\begin{bmatrix} 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix} \frac{\mathrm{d}\mathbf{c}}{\mathrm{d}t} = \begin{bmatrix} ([\mathrm{As(III)}]_0 + [\mathrm{As(V)}]_0) - ([\mathrm{As(III)}] + [\mathrm{As(III)}]_{\mathrm{ads}} + [\mathrm{As(V)}]_{\mathrm{tot}}) \\ [\mathrm{As(V)}]_{\mathrm{tot}} - ([\mathrm{As(V)}]_{\mathrm{ads}} + [\mathrm{As(V)}]) \\ [\mathrm{P}] - ([\mathrm{P}]_0 - [\mathrm{P}]_{\mathrm{ads}}) \\ [\mathrm{Si}] - ([\mathrm{Si}]_0 - [\mathrm{Si}]_{\mathrm{ads}}) \\ -k_{\mathrm{app}}[\mathrm{Fe(II)}][\mathrm{O_2}] + \mathrm{d}[\mathrm{Fe(II)}]_{\mathrm{dosage}}/\mathrm{d}t \\ k_{\mathrm{app}}[\mathrm{Fe(II)}][\mathrm{O_2}] \\ k_{\mathrm{r}}([\mathrm{O_2}]_{\mathrm{sat}} - [\mathrm{O_2}]) - k_{\mathrm{app}}[\mathrm{Fe(II)}][\mathrm{O_2}] \\ (\beta k_{\mathrm{app}}[\mathrm{Fe(II)}][\mathrm{O_2}])/(1 + ((k_1[\mathrm{Fe(II)}])/(k_2[\mathrm{As(III)}]))) \\ \mathrm{d}/\mathrm{d}[\mathrm{Fe(II)}]_{\mathrm{dosage}} t - k_{\mathrm{H_2O_2}}[\mathrm{Fe(II)}][\mathrm{H_2O_2}] - L(t) \\ 0 \end{bmatrix}$$

which is a stiff index 1 differential algebraic equation, solvable by regular methods.

3.2 Constants

The constants used in the equations are as follows. Some are actually functions of pH.

- $K_{\mathrm{As(III)}} \approx 10^{3.81}$
- $K_{\mathrm{P}}(\mathrm{pH}) \approx 10^{10.30 - 0.65 \cdot \mathrm{pH}}$
- $K_{\mathrm{As(V)}}(\mathrm{pH}) \approx 10^{8.95 - 0.53 \cdot \mathrm{pH}}$
- $K_{\mathrm{Si}}(\mathrm{pH}) \approx 10^{-6.63 + 1.15 \cdot \mathrm{pH}}$
- $K_{\mathrm{H_2O_2}}(\mathrm{pH}) \approx 10^{0.89 \cdot \mathrm{pH} - 2.55}$
- $k_{\mathrm{app}}(\mathrm{pH}) \approx 10^{-10.58 + 1.64 \cdot \mathrm{pH}}$
- $\frac{k_1}{k_2}(\mathrm{pH}) \approx -6.63 + 1.15 \cdot \mathrm{pH}$
- $k_{\mathrm{r}} \approx 10^{-3.26}\,\mathrm{s}^{-1}$
- $q_{\max}(\mathrm{pH}) \approx -2.19 + 0.45 \cdot \mathrm{pH}$
- $\beta \approx 0.25$

4 Variable Catalogue

4.1 Constants

- `faraday_constant`: $\mathcal{F} \approx 96\,485\,\mathrm{C\,mol^{-1}}$
- `mm_x`: molar mass of species x
- `O2_saturation_20C`: $[\mathrm{O_2}]_{\mathrm{sat}}$ at 20 °C
- `O2_saturation_25C`: $[\mathrm{O_2}]_{\mathrm{sat}}$ at 25 °C
- `K_AsIII`: $K_{\mathrm{As(III)}}$
- `K_AsV`: $K_{\mathrm{As(V)}}$ (function of pH)
- `K_P`: K_{P} (function of pH)
- `K_Si`: K_{Si} (function of pH)
- `K_H2O2`: $K_{\mathrm{H_2O_2}}$ (function of pH)
- `q_max`: $q_{\max}$ (function of pH)
- `k_app`: k_{app} (function of pH)
- `k_1_div_k_2`: k_1/k_2 (function of pH)
- `k_r`: k_{r}
- `beta`: β

4.2 Other Variables

- `dose_time`: time of dosage, minutes
- `dose_rate`: $\mathrm{d}[\mathrm{Fe(II)}]_{\mathrm{dosage}}/\mathrm{d}t$, coulombs per second
- `dose_load`: $[\mathrm{Fe(II)}]_{\mathrm{dosage}}$, coulombs
- `mix_time`: time of mixing (zero dosing), minutes
- `initial_conc`: initial concentration vector $\mathbf{c}(0)$, units are mM
- `fe_dosage_rate`: `dose_rate` in units of $\mathrm{M\,s^{-1}}$
- `fe_dosage_load`: `dose_load` in units of M
- `time_list`, `conc_list`: arrays holding time and concentration values from simulation (`conc_list[i]` $= \mathbf{c}($`time_dose[i]`$)$)
- `time_dose_list`, `conc_dose`: arrays holding time and concentration values from simulation during dosage period (`conc_dose[i]` $= \mathbf{c}($`time_dose[i]`$)$)

4.3 Vectors

- Concentration vector $\mathbf{c}$:

$$\mathbf{c} = \left(\underbrace{[\text{As(III)}]}_{\texttt{c[0]}}, \underbrace{[\text{As(V)}]}_{\texttt{c[1]}}, \underbrace{[\text{P}]}_{\texttt{c[2]}}, \underbrace{[\text{Si}]}_{\texttt{c[3]}}, \underbrace{[\text{Fe(II)}]}_{\texttt{c[4]}}, \underbrace{[\text{Fe(III)}]}_{\texttt{c[5]}}, \underbrace{[\text{O}_2]}_{\texttt{c[6]}}, \underbrace{[\text{As(V)}]_{\text{tot}}}_{\texttt{c[7]}}, \underbrace{\text{pH}}_{\texttt{c[8]}} \right)$$

- Adsorbed species vector $\mathbf{a}$:

$$\mathbf{a} = \left(\underbrace{[\text{As(III)}]_{\text{ads}}}_{\texttt{a[0]}}, \underbrace{[\text{As(V)}]_{\text{ads}}}_{\texttt{a[1]}}, \underbrace{[\text{P}]_{\text{ads}}}_{\texttt{a[2]}}, \underbrace{[\text{Si}]_{\text{ads}}}_{\texttt{a[3]}} \right)$$

5 Integration Implementation

We utilize an implicit problem model using the Assimulo library, integrated using. The residue function $\text{Res}(t; \mathbf{c}, \mathrm{d}\mathbf{c}/\mathrm{d}t)$ (which is simply one side of the differential algebraic equation subtracted from the other) is a function in both $\mathbf{c}$ and $\mathrm{d}\mathbf{c}/\mathrm{d}t$, so it is required to provide initial conditions for both $\mathbf{c}$ and its derivative. We concern ourselves only with the problem at $t = 0$, since otherwise we can integrate to the time value and take the values in $\mathbf{c}$ and $\mathrm{d}\mathbf{c}/\mathrm{d}t$ from there.

We have that the inital conditions in $\mathbf{c}$ are given by manual input. On the flip side, we compute that the initial conditions in $\mathrm{d}\mathbf{c}/\mathrm{d}t$ are

$$\left.\frac{\mathrm{d}\mathbf{c}}{\mathrm{d}t}\right|_{t=0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -k_{\text{app}}[\text{Fe(II)}]_0[\text{O}_2]_0 + \mathrm{d}[\text{Fe(II)}]_{\text{dosage}}/\mathrm{d}t \\ k_{\text{app}}[\text{Fe(II)}]_0[\text{O}_2]_0 \\ k_{\text{r}}([\text{O}_2]_{\text{sat}} - [\text{O}_2]_0) - k_{\text{app}}[\text{Fe(II)}]_0[\text{O}_2]_0 \\ (\beta k_{\text{app}}[\text{Fe(II)}]_0[\text{O}_2]_0)/(1 + ((k_1[\text{Fe(II)}]_0)/(k_2[\text{As(III)}]_0))) \\ 0 \end{bmatrix}$$

(see some computations above for reasoning).

These initial conditions are plugged into the Radau5 solver, with an absolute tolerance of 1 part in 10^7 and a relative tolerance of 1 part in 10^5.

The system is relatively stiff, which is accounted for by employing the `Radau5DAE` library, which is good for stiff problems.